

Compiling on Windows

Prerequisites

- for OpenCPN v5.0.0
- with wxWidgets v3.1.2
- [Plugin API](#) starts with API 1.16 (Supports plugins with earlier API but they must be compiled again for OpenCPN v5.0 due to the change in wxWidgets.)
- Highest chance of success if these steps are followed exactly.

Visual Studio 2017

- Get Visual Studio Community 2017. Go to this microsoft web page [Older Visual Studio](#) and scroll down.
- Install and select the workload **Desktop development with C++** and include component **Windows XP support for C++**.
- Also include the appropriate **Windows x SDK** component. Generally install the most current version, Windows 10 preferred.

Git

- Get Git for Windows from <https://gitforwindows.org>
- Install and let the installer register git in your PATH and select **Checkout as-is, commit Unix-style line endings**

CMake

- Get the latest CMake installation packages from <https://cmake.org>
- Install and let the installer register in the PATH environment variable

POedit

- Get the latest POedit installation package from <https://poedit.net>
- Install
- Add **C:\Program Files (x86)\Poedit\GettextTools\bin** to your PATH

NSIS

- Download NSIS (Nullsoft Scriptable Install System) from <https://nsis.sourceforge.io/Download>

- Install

7-Zip

- Download 7-Zip from <https://www.7-zip.org/download.html>
- Install

Get the wxWidgets 3.1.2 sources and build them

- Open a **x86 Native Tools Command Prompt for VS 2017** (In Windows 10 by typing in the search box next to the Windows start button **x86 Native...**etc) and execute the following commands:

```
cd C:\Users\myname\Sources
git clone -b v3.1.2 https://github.com/wxWidgets/wxWidgets
cd wxWidgets
git submodule init
git submodule update

cd build\msw
nmake /f makefile.vc BUILD=release SHARED=1 CXXFLAGS=/D_USING_V141_SDK71_
CFLAGS=/D_USING_V141_SDK71_ LDFLAGS=/SUBSYSTEM:WINDOWS,5.01
nmake /f makefile.vc BUILD=debug SHARED=1 CXXFLAGS=/D_USING_V141_SDK71_
CFLAGS=/D_USING_V141_SDK71_ LDFLAGS=/SUBSYSTEM:WINDOWS,5.01
```

- Set the environment variable **wxWIDGETS_ROOT_DIR** with a value of the top level directory of your wxWidgets source tree.

Alternatives:

- The commands on this page refers to C:\users\myname\Sources as the directory for all sources. Obviously one can use other directories.
- The commands above download the sources from GitHub and then build the release and the debug version. Instead of using git clone, you can download the wxWidgets sources manually from <https://github.com/wxWidgets/wxWidgets/releases/tag/v3.1.2>.
- Alternatively there is a faster way to get started, by just getting wxWidgets prebuilt from [wxWidgets-3.1.2.7z](#). This however is not suitable for serious development or debugging as only release configuration libraries are included.

Getting and building OpenCPN source

Get the OpenCPN sources

In the open **x86 Native Tools Command Prompt for VS 2017** execute the following commands:

```
cd \Users\myname\Sources
git clone https://github.com/OpenCPN/OpenCPN
```

Get prebuilt Windows dependencies

- Download the dependencies bundle from [OpenCPN_buildwin-4.99a.7z](#)
- Extract into **c:\Users\myname\Sources\OpenCPN\buildwin** using 7-Zip

Building OpenCPN

- In the open **x86 Native Tools Command Prompt for VS 2017** execute the following commands to create the build directory, generate the solutions files and build the debug version, the release version and the setup package.

```
cd C:\Users\myname\Sources\OpenCPN
mkdir build
cd build
cmake -G "Visual Studio 15 2017" -T v141_xp ..
cmake --build .
cmake --build . --config release
cmake --build . --config release --target package
```

- These commands should generate an OpenCPN install package in **c:\Users\username\Sources\OpenCPN\build\opencpn_5.0.0_setup.exe**
- Execute this program to install OpenCPN. Choose c:\Program Files (x86)\OpenCPN\ as installation directory to avoid unnecessary issues when installing plugins.

Alternatives:

- Instead of running the last 3 cmake-commands, one could also start Visual Studio, open the generated solutionfile OpenCPN.sln and build from there.
- Obviously other directories can be used as well, just as multiple opencpn installation. The instructions above are meant for those setting up a new development environment for just OpenCPN 5.0.

Setup Copyfiles.bat - Last step to Debug

Copyfiles.bat is a useful single batch file to copy all the needed files to the various directories. This batch file does not execute any cmake commands.

- Download [Copyfiles.bat.doc](#) and remove the .doc
- Move the file to <Your OpenCPN source tree> (for example: C:\Compile\Github\Opencpn)
- Execute copyfiles.bat
- Start Visual Studio 2017 and the OpenCPN-solution file. (For Example: C:\Compile\Github\Opencpn\build\opencpn.sln)
- If the Solution Explorer is not visible, open the Solution Explorer (Via the view-menu, or Ctrl+Alt+L)
- Select the project "Opencpn" from the list, right click and pick "Set as Startup Project"

Now Opencpn should be ready to debug.

BatchUTILS

[BatchUTILS](#) is another alternative that is a more complete set of batch files to assist building OpenCPN. Osetup.bat is intended to be modified by an individual user based on their own system. Git clone <https://github.com/transmitterdan/BatchUTILS> to your github directory, next to the OpenCPN local repository. Discussion about [the use and development of BatchUtils](#)

It's nearly impossible to guess all the possible wxWidgets versions someone might have on their system. Also, other tools sometimes change paths as new versions are released. So it is expected that each user will customize Osetup.bat. The latest of Transmitter Dan's batchutils is in git. If you want to use some other environment variable other than WXDIR that is ok. The Cmake tool will accept a number of possible environment names as the root of the wxWidgets toolkit.

First Run

1. Make sure you've downloaded wxWidgets 3.1.2 and compiled them See the paragraph above.
2. Be sure to set environment path for wxWIDGETS_ROOT_DIR
3. First try doing Release or RelWithDebInfo to avoid issues using Debug because of missing wxWidgets debug .dlls.
4. Do not use the "All Build" target. Use the "opencpn" target.
5. Right click on "opencpn" target, and select "Set as StartUp Project"
6. Build the target "opencpn". Should be able to debug this now.
7. If you get a popup similar to "this DLL can't be found; wxmsw312ud_gl_vc_custom.dll"
8. You will need to rebuild wxWidgets in Debug mode if you want to run the OCPN Debug build. Not difficult. See the paragraph above.
9. You can also get useful debugging from the "RelWithDebInfo" build.

From:
<https://opencpn.org/wiki/dokuwiki/> - **OpenCPN Manuals**

Permanent link:
https://opencpn.org/wiki/dokuwiki/doku.php?id=opencpn:developer_manual:developer_guide:compiling_windows

Last update: 2021/10/03 18:20

